Directional Real-Time Optimization Applied to a Kite-Control Simulation Benchmark

Directional Real-Time Optimization Applied to a Kite-Control Simulation Benchmark

Sean Costello* (*sean.costello@epfl.ch*), Grégory François* (*gregory.francois@epfl.ch*), Dominique Bonvin* (*dominique.bonvin@epfl.ch*).

* Laboratoire d'Automatique, Ecole Polytechnique Fédérale de Lausanne
CH-1015 Lausanne, Switzerland

***Abstract*— This paper applies a novel two-layer optimizing control scheme to a kite-control benchmark problem. The upper layer is a recent real-time optimization algorithm, called Directional Modifier Adaptation, which represents a variation of the popular Modifier Adaptation algorithm. The lower layer consists of a path-following controller that can follow arbitrary paths. Application to a challenging benchmark scenario in simulation shows that this two-layer scheme is capable of substantially improving the performance of a complex system affected by significant stochastic disturbances, measurement noise and plant-model mismatch, while respecting operational constraints.**

I. Introduction

Kite power has received growing attention over the past few years and is now seen as a promising renewable wind-power technology [1]. The open problem of controlling a power-producing kite during dynamic flight is currently of great relevance, as it is one of the main technological challenges facing kite power. In addition, like any other power technology, there is a clear need to optimize the performances of a flying kite, while enforcing the operational satisfaction of the constraints, the most obvious being to ensure that the kite flies sufficiently high and does not crash into the ground.

Optimization of kites is limited by several issues:

1) First, while most optimization methods are model-based, modeling a kite is not an easy task, and accurate dynamic models are generally too detailed to be used for optimization purposes.
2) Kite control is typically performed by defining a cyclic path that the kite will have to follow. While several solutions have been proposed and tested [2], [3], [4], [5], the problem of accurately and robustly controlling a power-producing kite during dynamic flight remains an active area of research.
3) The third limitation is related to the availability of optimization methods that perform well in the presence of uncertainty due to plant-model mismatch and disturbances. Flying kites are difficult to model. The wind, which is the disturbance, is even more difficult to model. Thus, uncertainty is invariably present.

In this paper, the two-layer optimizing control scheme developed in [6] is applied to a simulated tethered kite. Two models are used, one being the control *model* that is used for controller design and optimization purposes, while the second corresponds to the simulated reality (referred to as the *plant*) that is different from the available model. This way, uncertainty is present in the form of plant-model mismatch. The two models are proposed to the research community as a benchmark problem for control and optimization in the presence of uncertainty in [7]. Since the problem can be studied in simulation, it is easily accessible to researchers in control theory. This benchmark is designed to reproduce a number of the challenges found in real systems, namely unstable nonlinear dynamics, plant-model mismatch, measurement noise and stochastic disturbances. It is based on industrial data, experimental studies from the literature, and one of the authors' own experience in experimental kite control.

To face the second and third aforementioned limitations, the two-layer scheme proposed in this article consists of a novel Real-Time Optimization (RTO) technique combined with a path-following controller. The results demonstrate that this scheme is capable of achieving near-optimal operation for a complex, unstable system affected by significant disturbances.

The RTO algorithm presented here is described in a general setting, as it has much wider application than kite power. Indeed, RTO is widely applied in different forms across many industrial sectors. In practice, real processes have a certain number of degrees of freedom, the values of which are chosen by operators to meet safety requirements and operating constraints and to optimize performance. RTO aims to find the optimal values for these degrees of freedom, and then to continually update them in response to disturbances and process variation. The 'two-step' approach [8], [9] is the traditional RTO technique, in which parameter estimation and model-based optimization are repeated until convergence occurs. However, the model must satisfy 'model adequacy' conditions in order for the RTO scheme to converge to the plant optimum [10], [11], [12]. These conditions are extremely stringent and are unlikely to be satisfied in a practical setting. Indeed, recent investigations have shown that, in the presence of structural plant-model mismatch, parameter estimation and re-optimization is ineffective, and can even lead to worse performance than without RTO [13],

[14], [15].

An alternative technique is the Modifier Adaptation (MA) algorithm, which has been successfully applied to a number of reasonably complex industrially relevant systems, namely, an experimental solid-oxide fuel-cell stack [16], the simulated heat and power system of a sugar and ethanol plant [17], and a simulated oxygen consumption plant [18]. MA does not perform parameter estimation, but rather uses measurements to update the cost and constraint functions in the model-based optimization problem. This update typically requires an estimate of the *gradients* of the plant cost and constraints. In theory, the technique guarantees optimality for the plant upon convergence. Many aspects of MA have recently been investigated: extension to closed-loop systems [19], approaches for estimating (or avoiding estimating) gradients [20], [21], [22], [23], [24], extension to discontinuous systems [25], conditioning of the model to guarantee convergence to the plant optimum [26], second-order modifiers [27], and even a promising result on sufficient conditions for global convergence [28].

Gradient estimation is the main barrier to implementing MA on industrial-scale systems. If the process has many inputs (RTO degrees of freedom), a prohibitive number of measurements is required to estimate the plant gradients. This negatively impacts the convergence speed. This is particularly true in the case of flying kites, for which the RTO degrees of freedom define the *continuous* reference path followed by the kite. This path is often parametrized using a large number of parameters, which results in many gradient elements to be estimated. Hence, this paper outlines a novel RTO method called *Dual Directional Modifier Adaptation* (dual D-MA) that addresses this issue by focusing on gradient estimation only in certain *privileged directions* of the input space. The result is an algorithm with the following characteristics:

1) The scheme does not require an accurate model nor parameter estimation.
2) *Constraint satisfaction* is ensured upon convergence.
3) A directional *optimality guarantee* for the *plant* is provided upon convergence.
4) *Fast convergence* can be achieved, even in the presence of significant process noise.
5) The design of the RTO scheme is *straightforward* using the available model.

The paper is structured as follows: Section II presents the benchmark problem, Section III develops a novel path-following controller for kites, Section IV presents the novel dual D-MA RTO algorithm, and Section V discusses the application of RTO and path-following control to the benchmark system. Finally, Section VI concludes and discusses future research directions.

II. Kite-Control Simulation Benchmark

This section briefly describes the simulation benchmark, full details along with simulation files can be obtained from [7]. The system under investigation is a kite (wing) on a fixed-length tether, depicted in Figure 1.

As mentioned in the introduction, a simple model is available for control and optimization purposes. However the simulated *reality* is based on a different set of differential and algebraic equations that account for changes in the kite's angle-of-attack, wind variations with altitude, wind gusting due to turbulence (variations with time), and the dynamics of the steering actuator. These detailed equations are supposed to be unknown for control design and optimization. In addition, the simulator returns measurements that are contaminated by stochastic measurement noise.

The kite's position is described by the spherical co-ordinates φ and ϑ, while its orientation (similar to a yaw angle) is defined by ψ. The wind is approximately aligned with the x axis. [1] The kite can be steered left and right using an actuator. If the kite is steered such that it flies roughly perpendicular to the wind, it will reach many times the wind speed. This results in very large aerodynamic forces on the kite, which are transmitted to the ground via the tension in the tether. The tether tension can be exploited either to drive a generator for electricity generation [4], or to help propel a ship [29].

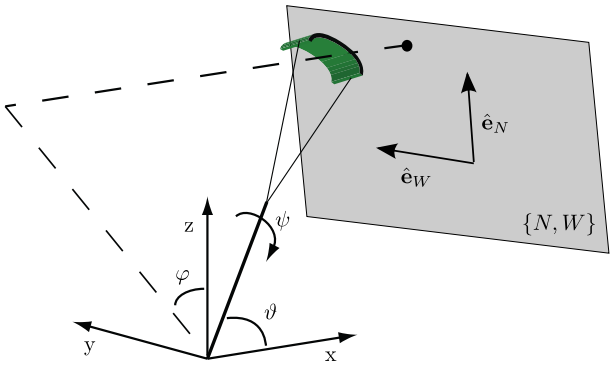

Fig. 1. Spherical co-ordinate system for the kite's position. The x and y axes are horizontal, while the z-axis points skywards. The kite is tethered to the origin. The kite's position can also be represented as a projection onto the $\{N, W\}$ plane.

A. Optimal control problem

The objective is to steer the kite such that the line tension, T, is maximized. As the kite will typically follow a roughly periodic path, and the line tension will vary during each cycle, the average line tension is considered. For a given time horizon, t_f, the problem can be formulated as:

$$\underset{\delta(t)}{\text{maximize}} \quad \bar{T}(t_\mathrm{f}) := \frac{1}{t_\mathrm{f}} \int_0^{t_\mathrm{f}} T(t)dt, \tag{II.1}$$

$$\text{subject to} \quad |\delta(t)| \leq \delta_{\max}, \tag{II.2}$$

$$r \sin\vartheta(t) \cos\varphi(t) \geq z_{\min}, \tag{II.3}$$

$$|\psi(t)| \leq 2\pi, \tag{II.4}$$

[1] A 2-D reference system for the kite is defined by the two orthogonal vectors $\hat{\mathbf{e}}_W = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T$ and $\hat{\mathbf{e}}_N = \begin{bmatrix} -\sin\bar{\vartheta} & 0 & \cos\bar{\vartheta} \end{bmatrix}^T$ (radians), which are tangent to the sphere upon which the kite can move, at the point $\{\bar{\vartheta}, \bar{\varphi}\} = \{0.3, 0\}$ rad. During normal flight, the kite remains close to this point.

where the manipulated variable $\delta(t)$ is the set-point for the steering actuator, with maximum value $\delta_{\max}$, r is the tether length, and $z_{\min}$ is the minimum allowable altitude. The last constraint is a winding constraint for the tether to ensure it is not twisted.

B. Control model

The model available for control design is a rough approximation of the plant. The dynamic equations are simple analytical expressions that help understand the kite's behavior (the interested reader is invited to see [2] for more insight into the model equations):

$$\dot{\vartheta} = \frac{w_{\rm ap}}{r}\left(\cos\psi - \frac{\tan\vartheta}{E}\right), \tag{II.5}$$

$$\dot{\varphi} = -\frac{w_{\rm ap}}{r\sin\vartheta}\sin\psi, \tag{II.6}$$

$$\dot{\psi} = w_{\rm ap} g_{\rm s}\delta + \dot{\varphi}\cos\vartheta, \tag{II.7}$$

where $g_{\rm s}$ is the turning constant. The lift/drag ratio, E, and the magnitude of the apparent wind projected onto the quarter sphere's tangent plane at the kite position, $w_{\rm ap}$, are given by

$$w_{\rm ap} = w_0 E\cos\vartheta, \tag{II.8}$$

$$E = E_0 - c\delta^2, \tag{II.9}$$

where w_0 is the wind speed, which is assumed constant in the model, and c is the turning penalty factor. The line tension is given by

$$T = \left(\frac{1}{2}\rho A w_0^2\right)\cos^2\vartheta(E+1)\sqrt{E^2+1}. \tag{II.10}$$

C. Comparison between the Model and the Plant

The optimal periodic path for the plant is shown in Figure 2 (attained when $t_f \to \infty$ in (II.1)). The optimal path that can be calculated using the control model is also shown. They are both horizontal figure-of-eight shapes. This is the shape typically used to maximize line tension, both for power generation and in kite-based sports. Nonetheless, the position and the shape of the two paths differ. This is due to the substantial parametric and structural mismatch between the model and the plant. For example, the wind direction in the simulated reality makes an angle of 15° with the x-axis, which results in a lateral offset between the two paths. Also, the wind speed in the simulated reality increases with altitude, while the model assumes it is constant with respect to altitude. This results in the plant optimal solution being higher than the model optimal solution. It is shown in [7] that the optimal average thrust for the plant is $\bar{T} = 39.61$ kN. It was found that the best performance that could be achieved with a relatively simple control scheme was $\bar{T} = 32.5$ kN, about 20% less than the optimal value.

The state measurements available in the simulated reality are contaminated by 2-10% measurement noise, and 25% stochastic wind-speed noise results in significant, unpredictable variations in the line tension and the kite's behavior from one cycle to the next (see [7] for a detailed specification).

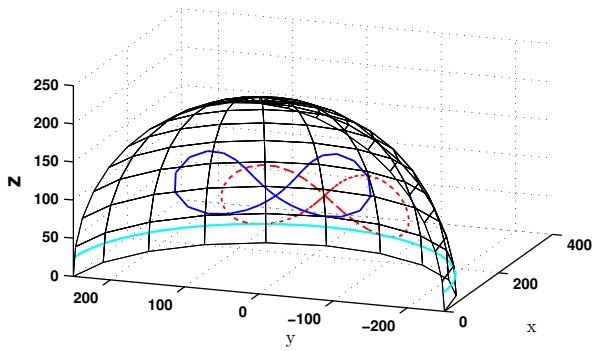

Fig. 2. Optimal path for the plant (blue) and for the model (red, dashed). The height constraint is shown in light blue.

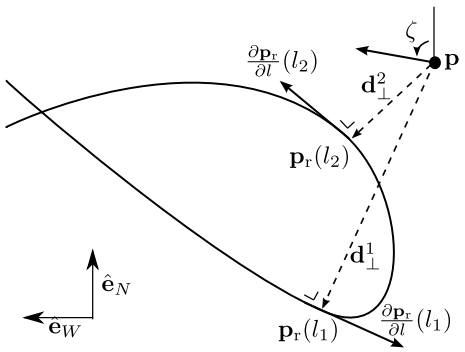

Fig. 3. Path-following controller: illustration of the kite's position relative to the reference path (all projected onto the $\{N, W\}$ plane, shown in Figure 1). $\mathbf{p}$ is the kite's position, and $\mathbf{p}_{\rm r}(l_1)$ and $\mathbf{p}_{\rm r}(l_2)$ are the two points on the path at which the path's tangent is perpendicular to the kite's position.

III. A PATH-FOLLOWING CONTROLLER FOR KITES

This section describes a simple 'vector-field' path-following controller for kites (or any vehicle moving on a plane). This type of controller is popular in the unmanned aerial-vehicle (UAV) community, where it was recently developed to follow circular and straight line paths (or composites of these) [30]. Here we extend it to arbitrary, smooth paths (including paths that intersect themselves). Firstly, the controller aims to control the position of the kite in the $\{N, W\}$ plane (shown in Figure 1). This position is denoted $\mathbf{p}$. As shown in Figure 3, we define the kite's velocity angle on this plane as:

$$\zeta = \angle\dot{\mathbf{p}} := \tan^{-1}\left(\frac{\dot{\mathbf{p}}_W}{\dot{\mathbf{p}}_N}\right). \tag{III.1}$$

This angle can be estimated from the measurements of the kite's position and orientation using linear transformations (note that during crosswind flight $\zeta \simeq -\psi$). The reference path on the $\{N, W\}$ plane is denoted $\mathbf{p}_{\rm r}(l)$, where l is the path length. The points on the path at which the path tangent is perpendicular to the kite position are $\mathbf{p}_{\rm r}(l_i)$. The angle of the path at each point is denoted as:

$$\zeta^i = \angle\frac{\partial \mathbf{p}_{\rm r}}{\partial l}(l_i), \tag{III.2}$$

while the vector pointing from the kite to each point is:

$$\mathbf{d}_{\perp}^{i} = \mathbf{p}_{\mathrm{r}}(l_i) - \mathbf{p}. \tag{III.3}$$

A *desired* heading angle corresponding to each point is obtained with the classic vector-field law [30]:

$$\begin{aligned}\zeta_{\mathrm{d}}^{i} =& \zeta^{i} + \zeta_{\mathrm{e}} \left(\frac{\|\mathbf{d}_{\perp}^{i}\|}{d_{\max}} \right)^{\beta} \\ &\times \mathrm{sgn}\left(\angle \left(\mathbf{d}_{\perp}^{i} - \frac{\partial \mathbf{p}_{\mathrm{r}}}{\partial l}(l_i) \right) \right) + \alpha \frac{\partial^2 \mathbf{p}_{\mathrm{r}}}{\partial l^2}(l_i),\end{aligned} \tag{III.4}$$

where the entry velocity angle ζ_{e} and the coefficient $\beta > 1$ are tuning parameters. The final term is a novel *curvature compensation* term that allows the controller to effectively follow a curved path. The curvature of the path indicates the rate of change of the path's angle (direction). Thanks to the curvature compensation, which can be varied by adjusting α, the controller anticipates curves in the path. Finally, the reference velocity angle is selected as the ζ_{d}^{i} that is closest to the kite's current velocity angle:

$$\zeta_{\mathrm{r}} = \zeta_{\mathrm{d}}^{i_{\mathrm{r}}}, \qquad i_{\mathrm{r}} = \underset{i}{\mathrm{argmin}} \quad |\zeta - \zeta_{\mathrm{d}}^{i}|. \tag{III.5}$$

A proportional control law is used to force the kite's velocity angle to follow the reference velocity angle:

$$\delta = K(\zeta_{\mathrm{r}} - \zeta). \tag{III.6}$$

IV. RTO METHOD: DUAL DIRECTIONAL MODIFIER ADAPTATION

This section presents the RTO method 'Dual Directional Modifier Adaptation' (dual D-MA) from [6]. It combines the concepts of directional derivatives, dual control, and statistically optimal gradient estimates with the existing Modifier-Adaptation RTO method.

A. RTO algorithm

The RTO algorithm aims to solve a problem of the form:

$$\begin{aligned}\mathbf{u}_{\mathrm{p}}^{*} :=& \arg\min_{\mathbf{u}} \ \phi_{\mathrm{p}}(\mathbf{u}) \\ \text{subject to} \quad & \mathbf{g}_{\mathrm{p}}(\mathbf{u}) \leq \mathbf{0},\end{aligned} \tag{IV.1}$$

where $\mathbf{u}$ is the n_u-dimensional vector of RTO inputs, ϕ_{p} the cost function and $\mathbf{g}_{\mathrm{p}}$ the n_g-dimensional vector of plant constraints. Here, the subscript $(\cdot)_{\mathrm{p}}$ indicates a quantity related to the plant, and we will refer to this as the plant optimization problem. The functions ϕ_{p} and $\mathbf{g}_{\mathrm{p}}$ are usually not known accurately, as only the models ϕ and $\mathbf{g}$ are available. Consequently, an approximate solution to the original problem (IV.1) is obtained by solving the following model-based problem:

$$\begin{aligned}\mathbf{u}^{*}(\boldsymbol{\theta}) :=& \arg\min_{\mathbf{u}} \ \phi(\mathbf{u}, \boldsymbol{\theta}) \\ \text{subject to} \quad & \mathbf{g}(\mathbf{u}, \boldsymbol{\theta}) \leq \mathbf{0},\end{aligned} \tag{IV.2}$$

where $\boldsymbol{\theta}$ is a n_θ-dimensional vector of uncertain model parameters. If the model matches the plant perfectly, solving Problem (IV.2) provides a solution to Problem (IV.1). Unfortunately, this is rarely the case since the structure of the model functions ϕ and $\mathbf{g}$ as well as the nominal values for the uncertain model parameters $\boldsymbol{\theta}_0$ are likely to be incorrect, which implies that the nominal model-based optimal input $\mathbf{u}^{*}(\boldsymbol{\theta}_0)$ will not correspond to $\mathbf{u}_{\mathrm{p}}^{*}$. As the plant optimal solution cannot be directly computed, it must be found through an intelligent iterative procedure. Different inputs $\mathbf{u}$ are applied successively to the plant and the resulting cost and constraints are measured. The measured cost and constraints for the input $\mathbf{u}_k$ are:

$$\tilde{\phi}_{\mathrm{p}}(\mathbf{u}_k) = \phi_{\mathrm{p}}(\mathbf{u}_k) + d_k^{\phi} \tag{IV.3}$$

$$\tilde{g}_{\mathrm{p},j}(\mathbf{u}_k) = g_{\mathrm{p},j}(\mathbf{u}_k) + d_k^{g,j}, \forall j \in [1, \ldots, n_g] \tag{IV.4}$$

where d_k^{ϕ} and $d_k^{g,j}$ are realizations of zero-mean random variables for the cost and the j^{th} constraint, respectively, with the corresponding variances σ_{ϕ}^2 and $\sigma_{g,j}^2$. This stochastic component represents high-frequency process noise. The measured values provide information about the plant, which, when appropriately combined with the (inaccurate) model, allows the plant optimum to be reached. The RTO algorithm treated here uses past data to estimate the derivatives of the plant cost and constraint functions in certain *privileged* directions. The cost and constraint gradients in the model-based optimization problem are then corrected, as are the constraint values.

The basic idea is to solve Problem (IV.2) with a modified cost function:

$$\phi_{\mathrm{m},k}(\mathbf{u}) := \phi(\mathbf{u}, \boldsymbol{\theta}_0) + (\boldsymbol{\lambda}_k^{\phi})^{T}(\mathbf{u} - \mathbf{u}_k), \tag{IV.5}$$

where $\mathbf{u}_k$ is the previous operating point, and $\boldsymbol{\lambda}_k^{\phi}$ is a first-order modifier term. The aim of this modifier is to locally match the gradient of the modified cost function with that of the plant cost function. Thus, ideally, the first-order modifier is determined such that [15]:

$$\left(\boldsymbol{\lambda}_k^{\phi}\right)^{T} = \nabla\phi_{\mathrm{p}}(\mathbf{u}_k) - \nabla\phi(\mathbf{u}_k, \boldsymbol{\theta}_0). \tag{IV.6}$$

However, this requires the gradient of the plant cost function, which is usually experimentally costly to calculate. A gradient estimate, $\nabla\phi_{\mathrm{E},k}$, is used in its place. This estimate does not aspire to be exact, i.e. $\nabla\phi_{\mathrm{E},k} \neq \nabla\phi_{\mathrm{p}}(\mathbf{u}_k)$. Instead, the estimate is required to closely approximate the plant gradient in a subspace, $\mathbf{U}_{\mathrm{r}}$, of *privileged* directions:

$$\nabla\phi_{\mathrm{E},k}\mathbf{U}_{\mathrm{r}} \simeq \nabla\phi_{\mathrm{p}}(\mathbf{u}_k)\mathbf{U}_{\mathrm{r}}. \tag{IV.7}$$

The gradient estimate is obtained from past operating data. It is a 'dual' algorithm because the RTO steps are designed not only to minimize the plant cost, but only to ensure an accurate gradient estimate can be obtained. To this purpose, the term $-c|\bar{\delta\mathbf{u}}^{T}(\mathbf{u} - \mathbf{u}_k)|^2$ is included in the modified cost function (c is a positive scalar) to encourage steps in the direction $\bar{\delta\mathbf{u}}$, the direction in which the variance of the current gradient estimate is greatest.

Algorithm 1: Dual Directional Modifier Adaptation

Initialize: Choose a matrix whose columns contain the privileged directions in the input space, $\mathbf{U}_{\mathrm{r}}$. Choose a positive

scalar *reward factor*, c_0, and set the initial reward coefficient $c = 0$. Initialize $\boldsymbol{\epsilon}_0 = \mathbf{0}$, $\boldsymbol{\lambda}_0^g = \mathbf{0}$, $\boldsymbol{\lambda}_0^\phi = \mathbf{0}$. Choose the modifier filter matrices $\mathbf{K}^\epsilon, \mathbf{K}^\mathrm{g}, \mathbf{K}^\phi$, typically diagonal matrices with eigenvalue in the interval $(0, 1]$. Initialize $\mathbf{u}_0$ with a conservative input (one that is unlikely to violate the plant constraints). Select values for $\Delta_{\max}$ and $\Delta^\mathrm{r}_{\max}$. Choose the tolerance for the gradient estimate variance in the privileged directions, σ^2_{TOL}, and set $\bar{\delta\mathbf{u}} = \mathbf{0}$.

for $k = 1 \to \infty$

1) Solve the modified model-based optimization problem

$$\begin{aligned} \mathbf{u}_k := \underset{\mathbf{u}}{\text{argmin}} \quad & \phi_{\mathrm{m},k-1}(\mathbf{u}) \\ \text{s.t.} \quad & \mathbf{g}_{\mathrm{m},k-1}(\mathbf{u}) \leq \mathbf{0}, \\ & \|\mathbf{u} - \mathbf{u}_{k-1}\| \leq \Delta_{max}, \end{aligned} \tag{IV.8}$$

where the modified cost and constraints are given by

$$\begin{aligned} \phi_{\mathrm{m},k}(\mathbf{u}) :=& \phi(\mathbf{u}, \boldsymbol{\theta}_0) + (\boldsymbol{\lambda}_k^\phi)^T(\mathbf{u} - \mathbf{u}_k) \\ & -c|\bar{\delta\mathbf{u}}^T(\mathbf{u} - \mathbf{u}_k)|^2, \end{aligned} \tag{IV.9}$$

$$\mathbf{g}_{\mathrm{m},k}(\mathbf{u}) := \mathbf{g}(\mathbf{u}, \boldsymbol{\theta}_0) + \boldsymbol{\epsilon}_k + (\boldsymbol{\lambda}_k^\mathrm{g})^T(\mathbf{u} - \mathbf{u}_k). \tag{IV.10}$$

2) Apply the input $\mathbf{u}_k$ to the plant, and observe the resulting cost and constraint values, $\tilde{\phi}_\mathrm{p}(\mathbf{u}_k)$ and $\tilde{\mathbf{g}}_\mathrm{p}(\mathbf{u}_k)$.
3) Use the gradient estimation algorithm from Section IV-C to estimate the plant cost gradient at the current operating point, $\nabla\phi_{\mathrm{E},k}$, and the gradient of each constraint, $\nabla g_{i,\mathrm{E},k}$, using past measurements. The algorithm will also calculate the variance of the cost gradient estimate, $\boldsymbol{\Sigma}^\phi_{\mathrm{E},k}$, and of *each* constraint gradient estimate, $\boldsymbol{\Sigma}^{g_j}_{\mathrm{E},k}\ \forall\ j = 1, \ldots, n_g$.
4) Get the direction in the column space of $\mathbf{U}_\mathrm{r}$ with maximum *Lagrangian*-estimate variance [2] :

$$\begin{aligned} \bar{\delta\mathbf{u}} \in \arg\max_{\delta\mathbf{u}} \quad & \delta\mathbf{u}^T\boldsymbol{\Sigma}^L_{\mathrm{E},k}\delta\mathbf{u} \\ s.t. \quad & \|\delta\mathbf{u}\| = 1, \\ & \delta\mathbf{u} \in C(\mathbf{U}_\mathrm{r}), \end{aligned} \tag{IV.11}$$

where $\boldsymbol{\Sigma}^L_{\mathrm{E},k} = \left(\boldsymbol{\Sigma}^\phi_{\mathrm{E},k} + \sum_{i=1}^{n_g} \nu_i \boldsymbol{\Sigma}^{g_i}_{\mathrm{E},k}\right)$ is the variance matrix of the Lagrangian gradient estimate ($\boldsymbol{\nu}$ is the Lagrange Multiplier obtained in Step 1).
5) **if** $\bar{\delta\mathbf{u}}^T\boldsymbol{\Sigma}^L_{\mathrm{E},k}\bar{\delta\mathbf{u}} > \sigma^2_{TOL}$
   $c = c_0$
   else
   $c = 0$
   end
6) Update the modifier terms using the measurements:

$$\begin{aligned} \boldsymbol{\epsilon}_k :=& (\mathbf{I}_{n_g} - \mathbf{K}^\epsilon)\boldsymbol{\epsilon}_{k-1} \\ & + \mathbf{K}^\epsilon\left(\tilde{\mathbf{g}}_\mathrm{p}(\mathbf{u}_k) - \mathbf{g}(\mathbf{u}_k, \boldsymbol{\theta}_0)\right), \end{aligned} \tag{IV.12}$$

$$\begin{aligned} \boldsymbol{\lambda}_k^\mathrm{g} :=& (\mathbf{I}_{n_u} - \mathbf{K}^\mathrm{g})\boldsymbol{\lambda}_{k-1}^\mathrm{g} \\ & + \mathbf{K}^\mathrm{g}\left(\nabla\mathbf{g}_{\mathrm{E},k} - \nabla\mathbf{g}(\mathbf{u}_k, \boldsymbol{\theta}_0)\right)^T, \end{aligned} \tag{IV.13}$$

$$\begin{aligned} \boldsymbol{\lambda}_k^\phi :=& (\mathbf{I}_{n_u} - \mathbf{K}^\phi)\boldsymbol{\lambda}_{k-1}^\phi \\ & + \mathbf{K}^\phi\left(\nabla\phi_{\mathrm{E},k} - \nabla\phi(\mathbf{u}_k, \boldsymbol{\theta}_0)\right)^T. \end{aligned} \tag{IV.14}$$

[2]Note that the solution to problem (IV.11) is the normalized dominant eigenvector of $\mathbf{U}_\mathrm{r}\mathbf{U}_\mathrm{r}^T\boldsymbol{\Sigma}^L_{\mathrm{E},k-1}\mathbf{U}_\mathrm{r}\mathbf{U}_\mathrm{r}^T$.

end

The algorithm's most important property is that, upon convergence, the plant performance cannot be improved by adapting the RTO input $\mathbf{u}$ *in the privileged directions*.

Theorem 4.1 (Directional Optimality upon Convergence): If the plant gradient estimates are accurate in the privileged directions (i.e. $\nabla\phi_{\mathrm{E},\infty}\mathbf{U}_\mathrm{r} = \nabla\phi_\mathrm{p}(\mathbf{u}_\infty)\mathbf{U}_\mathrm{r}$ and $\nabla\mathbf{g}_{\mathrm{E},\infty}\mathbf{U}_\mathrm{r} = \nabla\mathbf{g}_\mathrm{p}(\mathbf{u}_\infty)\mathbf{U}_\mathrm{r}$), then any point $\mathbf{u}_\infty$ that the Directional MA algorithm converges to is such that $\mathbf{r} = \mathbf{0}$ is a KKT point for the following problem:

$$\begin{aligned} \min_{\mathbf{r}} \quad & \phi_\mathrm{p}\left(\mathbf{u}_\infty + \mathbf{U}_\mathrm{r}\mathbf{r}\right) \\ \text{s.t.} \quad & \mathbf{g}_\mathrm{p}\left(\mathbf{u}_\infty + \mathbf{U}_\mathrm{r}\mathbf{r}\right) \leq \mathbf{0}\,. \end{aligned} \tag{IV.15}$$

Proof: An outline is given here, for the full proof see [6]. Any point the dual D-MA algorithm converges to is a KKT point for the modified model-based problem IV.8 (otherwise the algorithm would not stay there). This implies that $\exists\ \nu$ s.t. :

$$\nabla\phi_{\mathrm{E},k} + \boldsymbol{\nu}^T\nabla\mathbf{g}_{\mathrm{E},k} = \mathbf{0}. \tag{IV.16}$$

If it is assumed that the gradient estimates are perfect in the privileged directions, i.e. $\nabla\phi_{\mathrm{E},\infty}\mathbf{U}_\mathrm{r} = \nabla\phi_\mathrm{p}(\mathbf{u}_\infty)\mathbf{U}_\mathrm{r}$ (and likewise for the constraint gradient estimate), then

$$\left(\nabla\phi_\mathrm{p}(\mathbf{u}_\infty) + \boldsymbol{\nu}^T\nabla\mathbf{g}_\mathrm{p}(\mathbf{u}_\infty)\right)\mathbf{U}_\mathrm{r}\delta\mathbf{r} = 0. \tag{IV.17}$$

Hence, there is no (small) $\delta\mathbf{r}$ that can improve the plant cost without violating active constraints.

■

The algorithm's parameters can be tuned through simulation trials (alternatively, see [6] for an example of how to choose the algorithm's parameters in a methodological fashion). However, the matrix of privileged directions $\mathbf{U}_\mathrm{r}$ must be chosen with particular care.

B. Calculating the privileged directions

The most important aspect of D-MA is the choice of the privileged directions (columns of $\mathbf{U}_\mathrm{r}$). D-MA acts on two levels. It will a) adapt the input in the directions necessary to ensure constraint satisfaction, and b) try to minimize the cost by adapting the decision variable $\mathbf{u}$ in the privileged directions. It is important to note that, regardless of $\mathbf{U}_\mathrm{r}$, constraint satisfaction upon convergence is ensured. The dual D-MA algorithm attempts to match the Lagrangian's gradient for the modified model-based optimization problem with that of the plant-based problem. If this is achieved, then optimality for the plant is ensured. Parametric analysis of the model can be used to study the effect of parameter variations on the *Lagrangian's gradient*. If all likely parameter variations only cause notable change in the Lagrangian's gradient *in a few directions*, then it will suffice to only estimate gradients in these few directions. This is formalized in the following theorem.

Theorem 4.2 (Privileged Gradient Estimation Directions): In the event of small parametric plant/model mismatch, that is, $\phi_\mathrm{p}(\mathbf{u}) = \phi(\mathbf{u}, \boldsymbol{\theta}_\mathrm{p})$ and $\mathbf{g}_\mathrm{p}(\mathbf{u}) = \mathbf{g}(\mathbf{u}, \boldsymbol{\theta}_\mathrm{p})$ with

$\boldsymbol{\theta}_{\mathrm{p}} = \boldsymbol{\theta}_0 + \Delta\boldsymbol{\theta}$, the plant optimal solution $\mathbf{u}^*_{\mathrm{p}}$ is a fixed point for the D-MA algorithm if the direction matrix is chosen as:

$$\mathbf{U}_{\mathrm{r}} = \frac{\partial^2 \mathbf{L}}{\partial \mathbf{u} \partial \boldsymbol{\theta}}(\mathbf{u}^*(\boldsymbol{\theta}_0), \boldsymbol{\nu}^*(\boldsymbol{\theta}_0), \boldsymbol{\theta}_0) \quad \in \mathbb{R}^{n_u \times n_\theta}, \quad \text{(IV.18)}$$

where $\mathbf{L}(\mathbf{u}, \boldsymbol{\nu}, \boldsymbol{\theta}) = \phi(\mathbf{u}, \boldsymbol{\theta}) + \boldsymbol{\nu}^T \mathbf{g}(\mathbf{u}, \boldsymbol{\theta})$ is the Lagrangian, $\mathbf{u}^*(\boldsymbol{\theta}_0)$ is the nominal optimal solution, and $\boldsymbol{\nu}^*(\boldsymbol{\theta}_0)$ are the corresponding Lagrange multipliers for the model-based problem.

Proof: An outline is given here, for the full proof see [6]. The proof is based on the idea that the small parameter difference $\Delta\theta$ between the model and the plant will cause the following Lagrangian 'error':

$$\begin{aligned} &\Delta\boldsymbol{\theta}^T \frac{\partial^2 \mathbf{L}}{\partial \mathbf{u} \partial \boldsymbol{\theta}}^T (\mathbf{u}, \boldsymbol{\nu}, \boldsymbol{\theta}_0) \\ &= \frac{\partial \mathbf{L}}{\partial \mathbf{u}}(\mathbf{u}, \boldsymbol{\nu}, \boldsymbol{\theta}) - \frac{\partial \mathbf{L}}{\partial \mathbf{u}}(\mathbf{u}, \boldsymbol{\nu}, \boldsymbol{\theta}_0). \end{aligned} \quad \text{(IV.19)}$$

This gradient error belongs to the subspace defined by the columns of the matrix $\frac{\partial^2 \mathbf{L}}{\partial \mathbf{u} \partial \boldsymbol{\theta}}(\mathbf{u}, \boldsymbol{\nu}, \boldsymbol{\theta}_0)$. If the experimental gradient estimates are accurate for the directions spanning this subspace, the entirety of the Lagrangian's gradient error will be corrected for. Hence, the Lagrangian gradient for the modified model-based optimization problem IV.8 will match that of the plant optimization problem IV.1. Hence, optimality for the modified model-based problem also implies optimality for the plant.

■

Note that singular value decomposition(SVD) of the matrix $\frac{\partial^2 \mathbf{L}}{\partial \mathbf{u} \partial \boldsymbol{\theta}}(\mathbf{u}, \boldsymbol{\nu}, \boldsymbol{\theta}_0)$ can be used to further reduce the number of privileged directions by singling out those directions in which the Lagrangian's gradient will be *most* affected by parameter variations.

C. Gradient estimation

As dual D-MA is designed for RTO problems with many inputs, the past data points will generally neither be numerous enough, nor well enough distributed, to estimate the full plant gradient. Previously proposed methods for estimating the plant gradient from past data suffer from ill-conditioning in this situation [31], [22]. For this reason a novel gradient-estimation algorithm was proposed in [6].

The method is iterative. A reliable gradient estimate is constructed at each RTO iteration, starting with the nominal model gradient. The past measurements are integrated into the gradient estimate one at a time.

Algorithm 2: Iterative, weighted Broyden-update gradient estimator

Initialize: $\nabla\phi_{\mathrm{old}}$ and $\boldsymbol{\Sigma}_{\mathrm{old}}$ with the model gradient $\nabla\phi(\mathbf{u}_k, \boldsymbol{\theta}_0)$ and estimated model gradient covariance $\boldsymbol{\Sigma}^{\phi}_0$. Choose the radius around the current RTO point inside which past points will be used for gradient estimation, $\Delta^{\mathrm{r}}_{\max}$.

for $\forall\ j$ such that $\|\mathbf{u}_j - \mathbf{u}_k\| < \Delta^{\mathrm{r}}_{\max}$

1) $\delta\mathbf{u} = \frac{\mathbf{u}_j - \mathbf{u}_k}{\|\mathbf{u}_j - \mathbf{u}_k\|}$

2) Estimate the directional derivative in the $\delta\mathbf{u}$ direction:

$$\nabla_{\delta\mathbf{u}}\phi_{\mathrm{E}} = \frac{\tilde{\phi}_{\mathrm{p}}(\mathbf{u}_j) - \tilde{\phi}_{\mathrm{p}}(\mathbf{u}_k)}{\|\mathbf{u}_j - \mathbf{u}_k\|}. \quad \text{(IV.20)}$$

Also, calculate the variance of this estimate:

$$\sigma^2_{\mathrm{E}} = var\{\nabla_{\delta\mathbf{u}}\phi_{\mathrm{E}}\} = \frac{2\sigma^2_{\phi}}{\|\mathbf{u}_j - \mathbf{u}_k\|^2}. \quad \text{(IV.21)}$$

3) This estimate of the directional derivative can be combined with the existing gradient estimate, $\nabla\phi_{\mathrm{old}}$, using a weighted rank-1 (Broyden) update:

$$\begin{aligned} \nabla\phi_{\mathrm{new}} =& \nabla\phi_{\mathrm{old}} \\ &+\kappa(\nabla_{\delta\mathbf{u}}\phi_{\mathrm{E}} - \nabla\phi_{\mathrm{old}}\delta\mathbf{u})\delta\mathbf{u}^T, \end{aligned} \quad \text{(IV.22)}$$

where the variance matrix for the new gradient estimate is:

$$\begin{aligned} \boldsymbol{\Sigma}_{\mathrm{new}} =& (\mathbf{I} - \kappa\delta\mathbf{u}\delta\mathbf{u}^T)\boldsymbol{\Sigma}_{\mathrm{old}}(\mathbf{I} - \kappa\delta\mathbf{u}\delta\mathbf{u}^T) \\ &+\kappa^2\sigma^2_{\mathrm{E}}\delta\mathbf{u}\delta\mathbf{u}^T, \end{aligned} \quad \text{(IV.23)}$$

and

$$\kappa = \frac{\delta\mathbf{u}^T\boldsymbol{\Sigma}_{\mathrm{old}}\delta\mathbf{u}}{\delta\mathbf{u}^T\boldsymbol{\Sigma}_{\mathrm{old}}\delta\mathbf{u} + \sigma^2_E}. \quad \text{(IV.24)}$$

This value of κ minimizes $var\{\nabla\phi_{\mathrm{new}}\delta\mathbf{u}\} = \delta\mathbf{u}^T\boldsymbol{\Sigma}_{\mathrm{new}}\delta\mathbf{u}$ (see [6] for more details).

4) $\nabla\phi_{\mathrm{old}} = \nabla\phi_{\mathrm{new}}$ and $\boldsymbol{\Sigma}_{\mathrm{old}} = \boldsymbol{\Sigma}_{\mathrm{new}}$.

end

$\nabla\phi_{\mathrm{E},k} = \nabla\phi_{\mathrm{old}}$
$\boldsymbol{\Sigma}^{\phi}_{\mathrm{E},k} = \boldsymbol{\Sigma}_{\mathrm{old}}$

V. Application to the Kite-Control Benchmark

As shown in Figure 4, a two-layer scheme was applied to the benchmark. The lower layer consists of the path-following controller described in Section III that adjusts the steering-actuator set-point δ at the frequency of 8 Hz in order to follow the current periodic reference path $\mathbf{p}_{\mathrm{r},k}(l)$, where $l \in [0, 1]$ is the normalized path length. The upper RTO layer modifies this reference path each time the kite has completed one cycle of the current path. The RTO layer can be further subdivided into the dual D-MA algorithm from Section IV and a centering algorithm taken from [32]. The dual D-MA algorithm adjusts the *shape* and the height of the reference path, while the centering algorithm tries to align the center of the path (where it intersects itself) with the wind. By comparing the line tension during the right and left halves of the path, the centering algorithm calculates an angle by which the reference path must be rotated around the z axis, $\xi_{\mathrm{c},k}$ (see [32] for more details). Note that the dual D-MA algorithm could also be used to adjust the center of the path, however it is more efficient to use the centering algorithm, which makes clever use of the properties of this particular physical system.

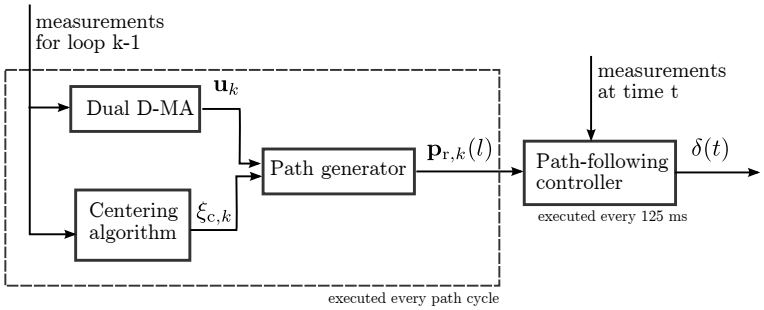

Fig. 4. Two-layer optimizing-control scheme for the kite-control benchmark.

The true RTO decision variable is the *continuous* reference *path* $\mathbf{p}_{\mathrm{r},k}(l)$. From a practical point of view, the RTO decision variable for dual D-MA is chosen as a finite set of points that parametrize the reference path:

$$\begin{aligned}\mathbf{u} =&[\vartheta_{\mathrm{r}}(0) \quad \varphi_{\mathrm{r}}(0) \quad \vartheta_{\mathrm{r}}(\tfrac{1}{N}) \quad \varphi_{\mathrm{r}}(\tfrac{1}{N}) \quad \vartheta_{\mathrm{r}}(\tfrac{2}{N}) \quad \varphi_{\mathrm{r}}(\tfrac{2}{N})\\ &\cdots \quad \vartheta_{\mathrm{r}}(\tfrac{N-1}{N}) \quad \varphi_{\mathrm{r}}(\tfrac{N-1}{N})]^T,\end{aligned} \tag{V.1}$$

where $N = n_u/2$ ($n_u = 40$ was used for this study). In the 'path generator' block, a continuous path, $\{\vartheta_{\mathrm{r}}(l), \varphi_{\mathrm{r}}(l)\}$, is obtained from $\mathbf{u}$ by fitting a spline to the points it contains. The spline is forced to be periodic (note that $\mathbf{u}$ does not specify the final point on the path), i.e. both the values and the slope of the spline at the endpoint must match:

$$\vartheta_{\mathrm{r}}(0) = \vartheta_{\mathrm{r}}(1), \qquad \varphi_{\mathrm{r}}(0) = \varphi_{\mathrm{r}}(1), \tag{V.2}$$

$$\dot{\vartheta}_{\mathrm{r}}(0) = \dot{\vartheta}_{\mathrm{r}}(1), \qquad \dot{\varphi}_{\mathrm{r}}(0) = \dot{\varphi}_{\mathrm{r}}(1). \tag{V.3}$$

The resulting continuous path is then rotated around the z-axis by the angle $\xi_{\mathrm{c},k}$ and projected onto the $\{N, W\}$ plane to yield the reference path $\mathbf{p}_{\mathrm{r},k}(l)$. The maximum steering-deflection constraint $|\delta| \leq \delta_{\max}$ is not taken into account by the RTO layer, as it is automatically enforced by the path-following controller. However, the height constraint is enforced by the RTO layer as, due to tracking error, the kite may consistently violate the path constraint even if the reference path does not. The constraint used in the model-based optimization problem IV.2 is formulated as:

$$\mathbf{g} = z_{\min} - \min_{l=0,\frac{1}{N},\ldots,\frac{N-1}{N}} r\sin\left(\vartheta_{\mathrm{r}}(l)\right)\cos\left(\varphi(l)\right), \tag{V.4}$$

while the plant constraint is defined as:

$$\mathbf{g}_{\mathrm{p}} = z_{\min} - \min_{t\in[0,t_{\mathrm{f}}]} r\sin(\vartheta(t))\cos\left(\varphi(t)\right). \tag{V.5}$$

The resulting discretized plant-optimization problem reads:

$$\begin{aligned}\mathbf{u}_{\mathrm{p}}^* =&\operatorname*{argmin}_{\mathbf{u}} \quad \phi_{\mathrm{p}}(\mathbf{u}) := -\bar{T}\\ \text{s.t.} \quad &\mathbf{g}_{\mathrm{p}}(\mathbf{u}) \leq \mathbf{0}.\end{aligned} \tag{V.6}$$

The benchmark system was simulated for 400 seconds. The initial reference path was chosen as the optimal path for the nominal model. In the dual D-MA algorithm, an initial constraint back-off of 25 m (the line length is 250 m) was used for the height constraint, i.e. $\epsilon_0 = 0.1$. The approach described in Section IV-B indicates that two privileged directions are sufficient. The initial reference path, along with variations of this path in the two privileged directions, are shown in Figure 5. Roughly speaking, the privileged directions influence the curvature and the height of the reference path.

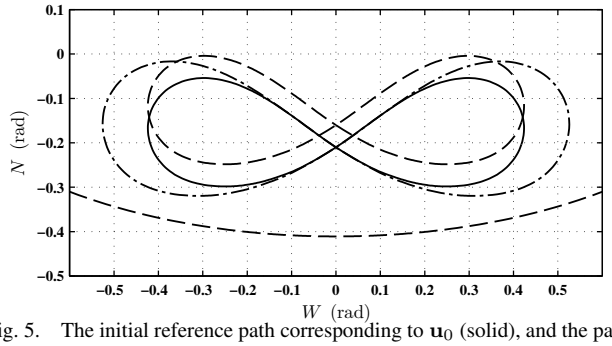

Fig. 5. The initial reference path corresponding to $\mathbf{u}_0$ (solid), and the paths corresponding to a small variation in each of the two privileged directions (dotted and dash-dotted).

Figure 6 shows the evolution of the line tension over time. It can be seen that the initial reference path is extremely sub-optimal. The RTO layer quickly adapts the path to nearly achieve the maximum possible line tension for the plant (about 10 cycles). The average line tension between 100 seconds and 400 seconds is 38.1 kN, about 4% less than the plant optimum. The kite's flight path is compared with the plant's optimal path in Figure 7. The pattern eventually followed by the kite is not identical to the plant's optimal path, however this difference results in negligible optimality loss. Note that the height constraint is very slightly violated. This is because the dual D-MA algorithm cannot robustly guarantee constraint satisfaction in the presence of process noise (which is significant in the benchmark scenario).

VI. Conclusion

This paper has presented the application of a complex 2-level optimizing-control scheme to a challenging Airborne Wind Energy simulation benchmark. The combination of the novel RTO algorithm Directional Modifier Adaptation with a vector-field path-following controller achieved very good performance. Nonetheless, ensuring robust constraint satisfaction in the presence of process disturbances remains an open issue.

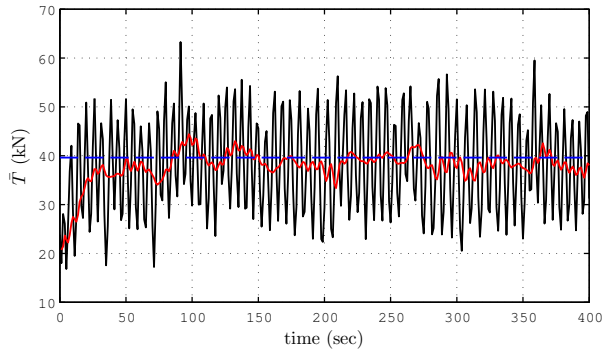

Fig. 6. Line tension vs. time (black), the average line tension over the past 10 seconds (red), and the maximum average line tension that can be achieved for the plant (blue).

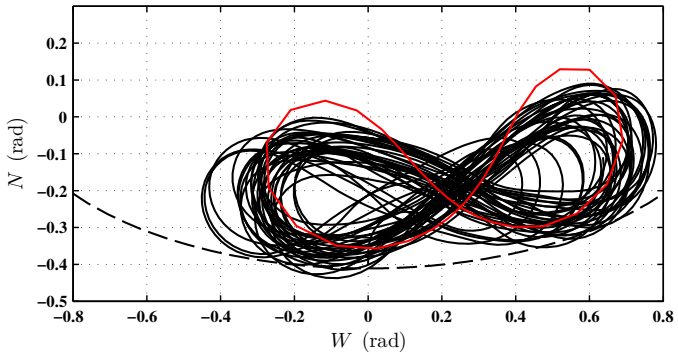

Fig. 7. The kite's flight path (black) and the plant optimal path (red). The height constraint is also shown (dashed).

REFERENCES

[1] U. Ahrens, M. Diehl, and R. Schmehl, Eds., *Airborne Wind Energy*. Berlin: Springer, 2013.

[2] M. Erhard and H. Strauch, "Control of towing kites for seagoing vessels," *IEEE Tran. on Control Systems Tech.*, vol. 21, no. 5, pp. 1629–1640, 2013.

[3] L. Fagiano, A. Zgraggen, M. Morari, and M. Khammash, "Automatic crosswind flight of tethered wings for airborne wind energy: Modeling, control design, and experimental results," *IEEE Tran. on Control Systems Tech.*, vol. 22, no. 4, pp. 1433–1447, 2014.

[4] R. Ruiterkamp and S. Sieberling, "Description and preliminary test results of a six degrees of freedom rigid wing pumping system," in *Airborne Wind Energy*, U. Ahrens, M. Diehl, and R. Schmehl, Eds. Berlin: Springer, 2013, pp. 443–458.

[5] C. Jehle and R. Schmehl, "Applied tracking control for kite power systems," *J. Guidance, Control, and Dynamics*, vol. 37, no. 4, pp. 1211–1222, 2014.

[6] S. Costello, G. François, and D. Bonvin, "A directional modifier adaptation algorithm for real-time optimization." *Submitted to J. Process Control*, 2014. [Online]. Available: http://infoscience.epfl.ch/record/202628

[7] ——, "Kite control - A benchmark problem for advanced control and dynamic optimization," *Submitted to Optimal Control Applications and Methods*, 2015.

[8] C. Y. Chen and B. Joseph, "On-line optimization using a two-phase approach: An application study," *Ind. Eng. Chem. Res.*, vol. 26, no. 9, pp. 1924–1930, 1987.

[9] S. Jang, B. Joseph, and H. Mukai, "On-line optimization of constrained multivariable chemical processes," *AIChE J.*, vol. 33, no. 1, pp. 26–35, 1987.

[10] J. Forbes, T. Marlin, and J. MacGregor, "Model adequacy requirements for optimizing plant operations," *Comp. Chem. Eng.*, vol. 18, no. 6, pp. 497–510, 1994.

[11] J. Forbes and T. Marlin, "Design cost: A systematic approach to technology selection for model-based real-time optimization systems," *Comp. Chem. Eng.*, vol. 20, no. 6-7, pp. 717–734, 1996.

[12] G. François and D. Bonvin, "Measurement-based real-time optimization of chemical processes," in *Advances in Chemical Engineering*, ser. Identification, Control and Optimisation of Proc. Sys. 43, S. Pushpavanam, Ed. Waltham: Academic Press, 2013, pp. 1–50.

[13] M. Agarwal, "Feasibility of on-line reoptimization in batch processes," *Chem. Eng. Communications*, vol. 158, no. 1, pp. 19–29, 1997.

[14] W. Gao and S. Engell, "Comparison of iterative set-point optimisation strategies under structural plant-model mismatch," in *Proc. IFAC World Congress*, vol. 16, 2005, pp. 401–401.

[15] A. G. Marchetti, "Modifier-adaptation methodology for real-time optimization," Ph.D. dissertation, # 4449, EPFL, Lausanne, 2009.

[16] G. A. Bunin, Z. Wuillemin, G. François, A. Nakajo, L. Tsikonis, and D. Bonvin, "Experimental real-time optimization of a solid oxide fuel cell stack via constraint adaptation," *Energy*, vol. 39, no. 1, pp. 54–62, 2012.

[17] F. J. Serralunga, M. C. Mussati, and P. A. Aguirre, "Model adaptation for real-time optimization in energy systems," *Ind. Eng. Chem. Res.*, vol. 52, no. 47, pp. 16 795–16 810, 2013.

[18] D. Navia, R. Martí, D. Sarabia, G. Gutirrez, and C. de Prada, "Handling infeasibilities in dual modifier-adaptation methodology for real-time optimization," in *Proc. 8th IFAC Symposium on Advanced Control of Chemical Processes*, 2012, pp. 537–542.

[19] S. Costello, G. François, D. Bonvin, and A. G. Marchetti, "Modifier adaptation for constrained closed-loop systems." in *Proc. IFAC World Congress*, vol. 19, 2014, pp. 11 080–11 086.

[20] G. A. Bunin, G. Francois, and D. Bonvin, "From discrete measurements to bounded gradient estimates: A look at some regularizing structures," *Ind. Eng. Chem. Res.*, vol. 52, no. 35, pp. 12 500–12 513, 2013.

[21] A. G. Marchetti, "A new dual modifier-adaptation approach for iterative process optimization with inaccurate models," *Comp. Chem. Eng.*, vol. 59, pp. 89–100, 2013.

[22] E. A. Rodger and B. Chachuat, "Design methodology of modifier adaptation for on-line optimization of uncertain processes," in *Proc. IFAC World Congress*, 2011, pp. 4113–4118.

[23] D. Navia, G. Gutiérrez, and C. de Prada, "Nested modifier-adaptation for RTO in the otto williams reactor," in *Proc. IFAC Symp. DYCOPS*, 2013, pp. 123–128.

[24] G. François and D. Bonvin, "Use of transient measurements for the optimization of steady-state performance via modifier adaptation," *Ind. Eng. Chem. Res.*, vol. 53, no. 13, pp. 5148–5159, 2014.

[25] F. J. Serralunga, P. A. Aguirre, and M. C. Mussati, "Including disjunctions in real-time optimization," *Ind. Eng. Chem. Res.*, vol. 53, no. 44, pp. 17 200–17 213, 2014.

[26] G. François and D. Bonvin, "Use of convex model approximations for real-time optimization via modifier adaptation," *Ind. Eng. Chem. Res.*, vol. 52, no. 33, pp. 11 614–11 625, 2013.

[27] T. Faulwasser and D. Bonvin, "On the use of second-order modifiers for real-time optimization," in *Proc. IFAC World Congress*, vol. 19, 2014.

[28] G. A. Bunin, "On the equivalence between the modifier-adaptation and trust-region frameworks," *Comp. Chem. Eng.*, vol. 71, pp. 154–157, 2014.

[29] F. Fritz, "Application of an automated kite system for ship propulsion and power generation," in *Airborne Wind Energy*, U. Ahrens, M. Diehl, and R. Schmehl, Eds. Berlin: Springer, 2013, pp. 359–372.

[30] D. Nelson, D. Barber, T. McLain, and R. Beard, "Vector field path following for miniature air vehicles," *IEEE Tran. on Robotics*, vol. 23, no. 3, pp. 519–529, 2007.

[31] A. Marchetti, B. Chachuat, and D. Bonvin, "A dual modifier-adaptation approach for real-time optimization," *J. Process Control*, vol. 20, no. 9, pp. 1027–1037, 2010.

[32] A. Zgraggen, L. Fagiano, and M. Morari, "Real-time optimization and adaptation of the crosswind flight of tethered wings for airborne wind energy," *IEEE Tran. on Control Systems Tech.*, vol. 23, no. 2, pp. 434–448, March 2015.